Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3

Lisandra V. Manzoni and Roberto T. Price, *Member*, *IEEE*

Abstract—This paper describes an assessment of the Rational Unified Process (RUP) based on the Capability Maturity Model (CMM). For each key practice (KP) identified in each key process area (KPA) of CMM levels 2 and 3, the Rational Unified Process was assessed to determine whether it satisfied the KP or not. For each KPA, the percentage of the key practices supported was calculated, and the results were tabulated. The report includes considerations about the coverage of each key process area, describing the highlights of the Rational Unified Process regarding its support for CMM levels 2 and 3, and suggests where an organization using it will need to complement it to conform to CMM. The assessment resulted in the elaboration of proposals to enhance the Rational Unified Process in order to satisfy the key process areas of CMM. Some of these are briefly described in this article.

Index Terms—Capability Maturity Model, CMM, process assessment model, Rational Unified Process, RUP, software process, software quality, Unified Modeling Language, UML.

1 Introduction

Researchers and practitioners have realized that processes cannot be defined and "frozen" once and for all [32]. Processes need to undergo changes and refinements continuously in order to increase their ability to deal with the requirements and expectations of all stakeholders, i.e., processes need to be continuously assessed and improved [32].

These observations have motivated the creation of quality models and standards for software process improvement [2], [3], [7], [10], [11], [17], [26]. One of the first and best known is the Capability Maturity Model (CMM) from the Carnegie Mellon University Software Engineering Institute (CMU/SEI) [16], [17], [18].

CMM provides a guide for selecting process improvement strategies by facilitating the determination of current process capabilities and the identification of the issues most critical to software quality and process improvement [16]. The first version of CMM was released in 1991 while the latest version (SW-CMM version 1.1), available at the SEI site, dates from 1993 [17]. CMM version 1.1 was used for this assessment. The Capability Maturity Model has evolved to Capability Maturity Model Integration (CMMI) [19], which enables the continued growth and expansion of the CMM concept to multiple disciplines, such as systems engineering, software engineering, integrated product and process development, and supplier sourcing.

The Rational Unified Process (RUP) [21] is a software engineering process that provides a disciplined approach to the assignment of tasks and responsibilities within a development organization [14]. Its goal is to ensure the production of high-quality software that meets the needs of its end users within a predictable schedule and budget [14], [21].

RUP uses the Unified Modeling Language (UML) [5] to model the software and describes how to apply best practices of software engineering via guidelines, templates, and tool mentors for all critical software lifecycle activities.

Within the growing number of text-books and articles on RUP, one finds descriptions such as "RUP is a software engineering process" [14], "RUP is a process framework" [14], "RUP is a process and a process framework" [14], and "RUP serves as the organization-wide process." [21] Both "process" as well as "process-framework" are seen at the same level in the metalevel hierarchy, the difference being that framework expresses the complete set of RUP activities from which a project-specific process or organization-wide process can be configured (essentially a tailored subset of RUP) [8], [21].

Being a development process, it is possible to assess RUP with respect to a reference model for process improvement such as CMM. This article describes such an applied assessment to RUP seen as a process framework. RUP is a proprietary product, sold as a CD or website access [21], which is essentially a hyperlink book. This paper refers to release 2001.03.00 of this product.

The option for CMM is because it is one of the best known models of software process maturity [6], [15]. It shows a detailed structure of software development process improvement and its practical usability has been

- *L.V. Manzoni is with the Serviço Federal de Processamento de Dados (SERPRO), RS, Brazil, and PhD student in computer science at the Universidade Federal do Rio Grande do Sul (UFRGS), RS, Brazil. E-mail: lisandra.manzoni@serpro.gov.br, lisandra@inf.ufrgs.br.*
- *R.T. Price is with the Instituto de Informática at the Universidade Federal do Rio Grande do Sul and with TeleNova Comms, 100 North Biscayne Boulevard Suite 2905, Miami, Florida 33132. E-mail: tomprice@inf.ufrgs.br, tomprice@telenova.net.*

Manuscript received 12 Nov. 2001; revised 26 Aug. 2002; accepted 16 Sept. 2002.
Recommended for acceptance by M. Shepperd.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 115357.

proven by several assessment cases [9]. CMM describes a process model that aims at helping an organization reach process maturity, besides being specific for software development, and strongly emphasizing aspects of continuous improvement [6].

The assessment was based on the key practices described by the Technical Report CMM/SEI-93-TR-025 Key Practices of the Capability Maturity Model [18]. For each key practice identified in this report, RUP was assessed to determine whether it describes elements that implement key practices or not. A key practice is considered RUP-supported when a substantial part (> 75 percent) of its subpractices are established.

This paper analyzes the Rational Unified Process by describing what it offers to support of CMM related activities and how and by identifying which aspects will require an organization using it to develop efforts to complement RUP to conform to CMM levels 2 and 3.

The result of the assessment is useful for organizations that use the Rational Unified Process because it highlights where RUP supports the implementation of the quality practices recommended by CMM, and where the process has to be enhanced to conform to CMM. The evaluation can also assist process engineers in the customization of RUP for specific projects, making it possible for an organization to deploy a subset of RUP, according to the key process areas it desires to reach.

The paper is outlined as follows: Section 2 briefly describes the Rational Unified Process. Section 3 shortly introduces the structure of the Capability Maturity Model. Section 4 describes the assessment conducted on RUP based on CMM, including the assessment process, a brief explanation of how RUP satisfies each key process area of levels 2 (repeatable process) and 3 (defined process), the summary of the results of the assessment. Finally, it describes some proposals to extend RUP. Section 5 mentions some related work and Section 6 concludes with observations regarding the results of the assessment and extensions needed by RUP to support all of CMM recommended key practices.

2 The Rational Unified Process

The Rational Unified Process is a software development process that describes the set of activities needed to transform user requirements into a software system. However, the Rational Unified Process is more than a single process; it is claimed to be a generic process framework that can be specialized for a large class of software systems, for different application areas, different types of organizations, different competence levels, and different project sizes [21].

RUP is a disciplined approach to assign and manage tasks and responsibilities in a development organization. The goal of this process is to produce, within a predictable schedule and budget, high-quality software that meets the needs of its end users [23].

The Rational Unified Process shows how a software development team can apply commercially proven approaches to software development [23]. These so-called "best practices," were selected by RUP designers not because they have precisely quantifiable values, but rather because successful organizations commonly use them in industry [14], [23]. These best practices are to develop software iteratively, to manage requirements, to use component-based architecture, to model software visually, to verify software quality, and to control changes to software.

The software lifecycle is based on four phases, through which the project evolves linearly in time. The phases are: Inception Phase, Elaboration Phase, Construction Phase, and Transaction Phase [23]. Each of these phases is composed of one or more iterations. Each iteration follows a waterfall pattern containing requirement elicitation, analysis, design, implementation, test, and deployment [21], [31], and results in a release (either internal or external) of an executable product, which grows incrementally from iteration to iteration to become the final system [21], [23]. Emphasis on these activities changes in different iterations and on different phases.

A model process describes who does what, how, and when [23]. The Rational Unified Process is represented with the use of four primary modeling elements, which are workers, activities, artifacts, and workflows [23].

A worker defines the behavior and responsibilities of an individual or a group of individuals working as a team. Workers represent a role played by individuals on a project, and define how they carry out their work. Workers have activities, which define the work they perform. An activity is something a worker does that provides a meaningful result in the context of the project.

Activities have input and output artifacts. An artifact is a work product of the process; workers use artifacts to perform activities, and produce artifacts in the execution of activities. Artifacts are pieces of information that are produced, modified, or used by a process.

Core workflows, in the Rational Unified Process, represent a partitioning of workers and activities into logical groupings called areas of concern or disciplines [14]. The core process workflows are grouped into six engineering workflows: Business Modeling, Requirements, Analysis and Design, Implementation, Test, and Deployment; and three supporting workflows: Project Management, Configuration and Change Management, and Environment [14].

3 Capability Maturity Model

The Capability Maturity Model for Software (CMM or SW-CMM), developed by the Software Engineering Institute (SEI), defines a five-level framework of how an organization matures its software development process [17]. CMM can be used for software process improvement, software process assessments, and software capability evaluations [17], [18].

In the following lines the structure of CMM is briefly described [17], [18].

CMM is composed of five maturity levels. A maturity level is a well-defined evolutionary plateau towards achieving a mature software process [18]. The maturity level of an organization software process indicates its process capabilities. The five levels are the following:

TABLE 1
Key Process Areas for Each Maturity Level

Maturity Level	Key Process Area
Repeatable (2)	Requirements Management
	Software Project Planning
	Software Project Tracking & Oversight
	Software Subcontract Management
	Software Quality Assurance
	Software Configuration Management
Defined (3)	Organization Process Focus
	Organization Process Definition
	Training Program
	Integrated Software Management
	Software Product Engineering
	Intergroup Coordination
	Peer Reviews
Managed (4)	Software Quality Management
	Quantitative Process Management
Optimizing (5)	Process Change Management
	Technology Change Management
	Defect Prevention

- **Initial.** The software process is characterized as ad hoc and occasionally even chaotic. Few processes are defined and success depends on individual effort and heroics.
- **Repeatable.** Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.
- **Defined.** Management and engineering activities are documented, standardized, and integrated into a family of standard software processes for the organization. Projects use a tailored version of the standard software processes of the organization for developing and maintaining software.
- **Managed.** Detailed measures of the software process and product quality are collected. Software processes and products are quantitatively understood and controlled.
- **Optimizing.** Continuous process improvement is facilitated by quantitative feedback from the process and from piloting innovative ideas and technologies.

With the exception of level 1, each maturity level is composed of several key process areas. Each key process area identifies a cluster of related activities (key practices) that, when performed collectively, achieve a set of goals considered important for establishing the process capability at that maturity level. Table 1 shows the key process areas for each maturity level.

Each key process area is described in terms of key practices that, when implemented, help to satisfy its goals. The key practices in each key process area are organized by a set of common features. The common features are attributes that indicate whether the implementation and institutionalization of a key process area are effective, repeatable, and lasting. The common features also group and order the key practices in a sequence that is helpful for organizations using them. The five common features are listed below [17]:

- **Commitment to Perform** describes the actions that the organization must take to ensure that the process is established and will endure. It involves establishing organizational policies and senior management sponsorship.
- **Ability to Perform** describes the preconditions that must exist in the project or organization to implement the software process competently. It involves resources, organizational structures, and training.
- **Activities Performed** describes the roles and procedures necessary to implement a key process area. It involves establishing plans and procedures, performing the work, tracking it, and taking corrective actions as necessary.
- **Measurement and Analysis** describes the need to measure the process, ways to do it and to analyze the measurements. It includes examples of measurements that could be taken to determine the status and effectiveness of the Activities Performed.
- **Verifying Implementation** describes the steps to ensure that the activities are performed in compliance with the process that has been established. It involves reviews and audits by management and software quality assurance.

The practices in the common feature Activities Performed describe what must be implemented to establish a process capability. The other practices, taken as a whole, form the basis through which an organization can institutionalize the practices described in the Activities Performed common feature [18].

The key practices of CMM were used for this assessment because they state the fundamental policies, procedures, and activities for the key process area [18]. The assessment was conducted by analyzing each of the 229 key practices described within the 13 key process areas of CMM levels 2 and 3.

Due to the large number of key practices, the summary identifying how RUP satisfies CMM, described in Section 4, is shown in groups of key process areas. A master's dissertation, available in its current Portuguese version at http://www.inf.ufrgs.br/amadeus/atuais/lisandra.html, has a detailed description of how RUP supports each CMM key practice (or why not) [33].

4 Using CMM to Assess RUP

Defining their own software process and assessing and improving it has been a matter of concern to many organizations that develop software, as demonstrated by the number of models of process assessment available [6], illustrated by models such as CMM [17], [18], Bootstrap [7], Trillium [2] and SQUID [3]; and the standardization efforts as the norms ISO9000-3 [10], ISO12207 [11], and ISO15504 (or SPICE) [26].

The key practices of CMM are expressed in terms of what is expected to be the normal practice of organizations that work on large government contracts. It is an SEI recommendation that, in any context in which CMM is

applied, a reasonable interpretation of how the practices should be applied must be used [18]. Technical Report CMM/SEI-94-TR-024 [29] contains guidelines on interpreting CMM.

This paper summarizes the assessment conducted by the authors to identify the RUP-provided support to organizations that want to reach CMM maturity levels 2 and 3. The objective of the assessment is to identify how RUP satisfies CMM key practices, and to identify practices not supported by RUP. The practices not supported by RUP are of great concern, as the organization should achieve them by others means.

The assessment used the key practices described by Technical Report CMM/SEI-93-TR-025 "Key Practices of the Capability Maturity Model" [18], Version 1.1 and the Rational Unified Process 2001.03.00, distributed by Rational Corporation [21].

The assessment was based on guidelines on how to interpret the model, identifying in RUP the elements described by CMM, regardless of the names used, matching those in RUP that fulfilled the proposed objectives of those in CMM. For example, the common feature of CMM Verifying Implementation contains practices that detail the review of activities by senior managers, while in RUP this activity is executed by the Project Reviewer. The objective is fulfilled, despite the use of different names for the activities and the roles of the workers that perform them.

In this assessment, all key practices of levels 2 and 3 are considered applicable. For some organizations or projects, some key practices may not be applicable; in which case, the particular recommendations that follow from the assessment may be ignored. For example, all key practices of Subcontract Management may be ignored if the organization does not practice any sort of outsourcing. The assessment process is sketched in the next section.

4.1 The Assessment Process

The assessment process followed a precise routine. For each key practice identified in the CMM report, the Rational Unified Process was assessed to determine whether it satisfied it or not. A CMM key practice is considered to be satisfied if there is a set of activities, artifacts, workers, or workflows in RUP to implement it.

For example, CMM describes the following key practice: The software engineering group uses the allocated requirements[1] as the basis for software plans, work products, and activities.

RUP satisfies this key practice because its Requirements Workflow describes a systematic approach to find, document, organize, and manage changes in software requirements. Software requirements are documented by means of Use Cases, and the Use Cases are used as the basic element for planning and monitoring the project.

In this assessment, a key practice is considered supported when a substantial part (>75 percent) of the subpractices associated with it are established by RUP. Subpractices, also known as subordinate key practices, are listed beneath the top-level key practices and describe what one would expect to find implemented for the top-level key practice [18].

The CMM key practices not supported by RUP were also identified. Any organization using RUP and aiming at CMM must look for alternatives to satisfy them.

For every key process area, a percentage of key practices satisfied was calculated. The analysis also considered the key practices grouped by common features.

The following section briefly describes how the Rational Unified Process supports each key process area (KPA) of CMM levels 2 and 3. This is done by stating how the elements described in RUP, such as workflows, activities, workers, and artifacts, support each CMM key process area.

4.2 Level 2—Repeatable Process

KPA: Requirements Management. The *Requirements Workflow* of RUP describes the sequence of activities that must be carried out to manage the requirements of the system and to execute changes to these requirements.

The requirements of the system are documented in the artifact *Software Requirements Specification*, which consists of a package containing *Use Cases* of the *Use Case Model* and applicable *Supplementary Specifications*, and the nontechnical requirements are documented in the artifacts *Stakeholder Requests* and *Vision*. RUP describes the measures related to the requirements in the artifact *Measurement Plan*. The artifact *Requirements Attributes* contains a repository of project requirements, attributes, and dependencies to track the requirements.

The worker *Change Control Manager* is responsible for approving and revising the change proposals in the requirements. The *Requirements Reviewer* plans and conducts the formal review of software requirements. The *System Analyst* leads and coordinates requirements elicitation and use-case modeling.

KPA: Software Project Planning. The *Project Management Workflow* of RUP describes the activities to manage software projects, such as: *Develop Software Development Plan*, *Plan Phases and Iterations*, *Project Planning Review*, and *Define Monitoring and Control Processes*. RUP proposes an iterative and incremental lifecycle.

Artifact *Software Development Plan* (SDP) describes the planning of the project; *Vision* describes the general vision of the requirements of the design. *Business Case* describes an economic analysis of the project. *Risk Management Plan* details how to manage the risks associated with the project. *Development Case* describes the infrastructure plans and tools. *Status Assessment* provides a mechanism for addressing, communicating, and resolving management issues, technical issues, and project risks. The project data are stored in artifact *Project Measurements*.

Project Manager is responsible for negotiating commitments and developing the SDP. The *Project Reviewer* is responsible for evaluating project planning and project assessment artifacts.

KPA: Software Project Tracking and Oversight. *Project Management Workflow* describes the activities that must be executed to tracking and oversight of the project, such as *Monitor and Control Project*, *Project Planning Review*, and to perform formal reviews at the end of each phase (*major*

1. The system requirements allocated to the software, usually referred to as the "allocated requirements" in CMM, are the subset of the system requirements that the software components of the system implement.

milestones). *Monitor and Control Project* has the main purpose of continuously monitoring the project in terms of active risks and objective measurements of progress and quality.

Project Manager is responsible for negotiating commitments to develop the SDP and to accomplish the project.

Iteration Plan describes a time-sequenced set of activities and tasks, with assigned resources, containing task dependencies for every iteration. *Risk List* enumerates known risks associated with specific mitigation or contingency. The project accomplishment data are stored in the artifact *Project Measurements*.

KPA: Software Subcontract Management. RUP does not support this key process area.

KPA: Software Quality Assurance. RUP proposes the elaboration of a *Quality Assurance Plan* for each project. This plan provides a clear view of how product, artifact, and process quality are to be assured.

The procedure to develop this plan is described in the activity *Develop Quality Assurance Plan* of the *Project Management Workflow*.

RUP recommends that the *Software Engineering Process Authority* (SEPA) should have responsibility for the process aspects of quality and perform process reviews and audits, as well as ensuring proper planning and conduct of the review events described in the *Review and Audit* section of the *Quality Assurance Plan*.

RUP is not clear about how to deal with the problems identified in the reviews and audits, neither does it state how the results of these reviews are communicated to the *Software Engineering Group* or to the workers.

KPA: Software Configuration Management. The *Configuration and Change Management Workflow* describes activities to manage configuration and to control change, such as *Establish Configuration Management Policies*, *Establish Change Control Process*, *Write Configuration Management Plan*, *Manage Baselines and Releases*, *Perform Configuration Audit and Monitor*, and *Report Configuration Status*.

Configuration Management Plan describes all Configuration and Change Control Management (CCM) activities that will be performed during the course of the product or project lifecycle.

Change Control Manager reviews submitted change requests to determine if it is a valid request or not. *Configuration Manager* provides the overall Configuration Management (CM) infrastructure and environment to the product development team. He is responsible for writing *CM Plan* and reporting progress statistics based on change requests. *Integrator* combines components to produce a build.

4.3 Level 3—Defined Process

KPA: Organization Process Focus. The activity *Assess Current Organization* in *Environment Workflow* describes the current status of the software organization in terms of its current process, tools, staff abilities and attitudes, customers, competitors, technical trends, problems, and improvement areas.

Software Engineering Process Authority (SEPA) facilitates the exchange of information and provides process guidance both to and from project practitioners, maintaining a current assessment of the organization's process maturity. *Process Engineer* is responsible for defining the software development process for each project.

RUP does not describe how the improvements identified at the process evaluation are implemented, neither does it include information about how to elaborate or review plans, as is proposed in CMM.

KPA: Organization Process Definition. RUP proposes in *Environment Workflow* the activities and the guidelines to configure the Rational Unified Process for a standard process of the organization and to configure the organization's standard process for a specific project. In many cases, the Rational Unified Process serves as the organization-wide process.

Development Case describes the configured development process for a given project. Organization's software process must be registered in a site, allowing access to all the staff organization.

Software Engineering Process Authority has the function to develop and to maintain the organization's process standard. The *Process Engineer* is responsible for the software development process itself, and to develop the *Development Case*.

The activities to define the process are scheduled in the *Software Development Plan* and they are accomplished in the activity *Monitor Project Status* of the *Project Management Workflow*.

KPA: Training Program. RUP does not support this key process area.

KPA: Integrated Software Management. The *Environment Workflow* specifies how to configure the organization's software process to a specific project. This configured process is described in the artifact *Development Case* and must be used to plan and to manage the project.

The activity *Project Planning Review* has the purpose of reviewing the *Software Development Plan* and the *Development Case*.

The data of the repository *Project Measurements* are used in the activities: *Monitor Project Status* and *Report Status*. In the *Plan Phases and Iterations* activity, RUP suggests that the collected data be used in future estimates.

The major milestones serve to review the state of the project at the end of a phase and determine whether the project should proceed to the next phase.

Risk Management Plan specifies how to manage the risks associated with a project, and *Risk List* identifies project risks, associating them with specific mitigation or contingency actions.

KPA: Software Product Engineering. Software engineering activities are described in the Rational Unified Process, through six process workflows, *Business Modeling, Requirements, Analysis and Design, Implementation, Test*, and *Deployment*; and three support workflows, *Configuration and Change Management, Project Management*, and *Environment*. RUP describes the relationships between the workflows and between the activities of each workflow. For each activity, RUP also describes the related artifacts and the workers responsible for the activity.

KPA: Intergroup Coordination. RUP describes the software engineering group and other engineering groups as *workers* of the process that relate with one another by means

TABLE 2
CMM Key Practices Coverage for RUP by Key Process Areas

Key Process Areas	Key Practices		
	Total	Supported	%
Requirements Management	12	10	83%
Software Project Planning	25	20	80%
Software Project Tracking & Oversight	24	20	83%
Software Subcontract Management	22	0	0%
Software Quality Assurance	17	10	59%
Software Configuration Management	21	18	86%
Organization Process Focus	16	7	44%
Organization Process Definition	11	9	82%
Training Program	16	0	0%
Integrated Software Management	19	15	79%
Software Product Engineering	20	16	80%
Intergroup Coordination	17	13	76%
Peer Reviews	9	6	67%

of the *artifacts* that they produce and of the activities that they execute.

Software Development Plan describes all the activities related to the software project. *Project Manager* has the responsibility to manage critical dependencies between the activities. Reviews and audits are held at some points of the development process. *Software Engineering Process Authority* has the purpose of facilitating the exchange of information between the participants of the process.

It is not clear in RUP how the activities and commitments at the system level are coordinated.

KPA: Peer Reviews. The reviews that will be held at the project are described in the *Review and Audit Plan* contained in the *Quality Assurance Plan*. Associated to the plan are guides on how to carry out reviews.

Review Record elaborated at the end of each review captures the results of the review of a project artifact, including review result, defects or problems identified, schedule, participants, etc.

4.4 Results of the Assessment

The results of the assessment are described in Tables 2, 3, 4, and 5. Table 2 shows the coverage of the CMM key practices by RUP, expressed in percentage of key practices, grouped by key process areas. Table 3 lists the missing or incomplete key practices. Table 4 shows the percentage of key practices satisfied by common features in each key process area. Table 5 shows the percentage of key practices supported by CMM, ignoring the common feature *Ability to Perform*. This common feature is eliminated because it describes the preconditions that must exist in the project or organization, such as resources, organizational structures, and training; issues considered beyond the scope of RUP.

For example, the key process area *Requirements Management* recommends 12 key practices that are divided among five common features. Table 2 shows that 10 of the 12 key practices are supported. Table 3 explicitly shows those key practices not supported by RUP. Table 4 shows how the key practices supported are distributed through the five common features (*Commitment to Perform, Ability to Perform, Activities Performed, Measurement and Analysis, Verifying Implementation*). Table 5 shows that, excluding the key practices of the common feature Ability to Perform, this key process area supports all eight key practices.

4.4.1 Key Practices Coverage

Table 2 shows that RUP establishes procedures that satisfy almost all the key practices described by CMM to the key process areas *Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Configuration Management, Organization Process Definition*, and *Software Product Engineering*.

In general, the missing key practices result from RUP neither describing how adequate human resources and funding are provided to carry out the process activities, nor ensuring that the development workers are trained to perform their activities.

The Requirements core workflow in RUP describes a sequence of activities that must be carried through to manage the software requirements and to provide changes in these requirements.

The activities related to the *Software Project Planning* and *Software Project Tracking and Oversight* are described in the Project Management Workflow.

RUP is limited to the planning of the software project, not describing the relationships of the software project with the overall project. Overall project planning involves the activities executed by other groups, as network infrastructure, hardware, marketing, etc. RUP does not describe how to identify, estimate, and track critical computer resources such as: computer memory capacity, computer processor usage, and communication channels capacity, etc.

In the Configuration and Change Management Workflow, the activities related to the key process area *Software Configuration Management* (SCM) are described.

Organization Process Definition involves developing and maintaining the standard software process of the organization, along with related process assets, such as description of software life cycles, process tailoring guidelines and criteria, and the support documents [18]. RUP describes guidelines to customize the Rational Unified Process for the process standard of the organization, considering the application domain, reuse practices, and core technologies mastered by the user organization. In the Environment Workflow, these guidelines to customize and to maintain the process are described.

Software Product Engineering describes the engineering tasks to build and to maintain the software. In RUP, these activities are described, mainly, in the process workflows: Business Modeling, Requirements, Analysis and Design, Implementation, Test, and Deployment.

RUP offers a good support to the key process area *Integrated Software Management*, and *Intergroup Coordination*. In these cases, it satisfies almost 70 percent of CMM-recommended key practices. *Integrated Software Management* describes how to use the standard software process, developed in the key process area *Organization Process*

TABLE 3
Missing or Incomplete CMM Key Practices

Key Process Areas	Missing or Incomplete Key Practices
Requirements Management	Adequate resources and funding are provided for managing the allocated requirements. The Software Engineering Group (SEG) is trained to perform their requirements manag. activities.
Software Project Planning	Adequate resources and funding are provided for planning the software project. Individuals involved in the project planning are trained in the estimating and planning procedures. Software project planning is initiated in the early stages of, and in parallel with, the overall project planning. The SEG participates with other affected groups in the overall project planning. Estimates for critical computer resources are derived.
Software Project Tracking & Oversight	Adequate resources and funding are provided for tracking the software project. Software managers (SoftMangs) are trained in technical and personnel aspects. First-line SoftMangs receive orientation in the technical aspects of the software project. Critical computer resources are tracked, and corrective actions are taken as necessary.
Software Subcontract Management	All
Software Quality Assurance	Adequate resources and funding are provided for performing the SQA activities. Members of the SQAG are trained to perform their SQA activities. Members of the software project receive orientation about the SQAG. SQAG participates in the preparation and review of the project software development, standards and procedures. The SQAG conducts periodic reviews of its activities and findings with the customer's SQA personnel. Measurements are made and used to determine the cost and schedule status of the SQA activities. Experts independent of the SQAG periodically review the activities and software work products of the project's SQA group.
Software Configuration Management	Adequate resources and funding are provided for performing SCM activities. Members of the SCM group are trained to perform SCM activities. Members of the SEG are trained to perform SCM activities.
Organization Process Focus	The organization follows a written organizational policy for coordinating software process development and improvement activities. Senior management oversees the organization's activities for software process development and improvement. Adequate resources and funding are provided for the organization's software process activities. Members of the group responsible for the organization's software process activities receive required training. Members of the SEG receive orientation on the organization's software process activities. The organization develops and maintains a plan for its software process development and improvement activities. Training for the organization's and projects' software processes is coordinated across the organization. Measurements are made and used to determine the status of the organization's process development and improvement activities. The activities for software process development and improvement are reviewed with senior manag.
Organization Process Definition	Adequate resources and funding are provided for developing and maintaining the software process. The individuals who develop and maintain the software process receive required training to perform these activities.
Training Program	All
Integrated Software Management	Adequate resources and funding are provided for managing the software project using the software process. The individuals responsible for developing the project's defined software process receive required training. The software managers receive required training. The project's critical computer resources are managed according to a documented procedure.
Software Product Engineering	Adequate resources and funding are provided for performing the software engineering tasks. Members of the SEG receive required training. Members of the SEG receive orientation in related software engineering disciplines. The project manager and all software managers receive orientation in the technical aspects of the software project.
Intergroup Coordination	Adequate resources and funding are provided for coordinating the software engineering activities with other engineering groups. Managers in the organization receive training in teamwork. All task leaders in each engineering group receive orientation in the processes, methods, etc. The members of the engineering groups receive orientation in working as a team.
Peer Reviews	Adequate resources and funding are provided for performing peer reviews. Peer review leaders receive required training in how to lead peer reviews. Reviewers receive required training in the objectives, principles, and methods of peer reviews.

Definition, in order to create a specific process for a project. RUP describes, in the Environment Workflow, procedures to customize the process for an organization (it can be the Rational Unified Process itself), for a specific project, and how this process must be documented in the Development Case. *Intergroup Coordination* involves establishing means

TABLE 4
Percentage of Key Practices Satisfied by RUP, Organized by Common Features

Key Process Areas	Common Features				
	CM	AB	AC	MA	VI
Requirements Management	100%	50%	100%	100%	100%
Software Project Planning	100%	50%	80,0%	100%	100%
Software Project Tracking & Oversight	100%	40%	92%	100%	100%
Software Quality Assurance	100%	25%	75%	0%	66%
Software Configuration Management	100%	40%	100%	100%	100%
Organization Process Focus	33%	25%	71%	0%	0%
Organization Process Definition	100%	0%	100%	100%	100%
Integrated Software Management	100%	0%	90%	100%	100%
Software Product Engineering	100%	0%	100%	100%	100%
Intergroup Coordination	100%	20%	100%	100%	100%
Peer Reviews	100%	0%	100%	100%	100%

TABLE 5
CMM Key Practices Coverage for RUP by Key Process Areas

Key Process Areas	Key Practices		
	Total	Supported	%
Requirements Management	8	8	100%
Software Project Planning	21	18	86%
Software Project Tracking & Oversight	19	18	95%
Software Quality Assurance	13	9	69%
Software Configuration Management	16	16	100%
Organization Process Focus	12	6	50%
Organization Process Definition	9	9	100%
Integrated Software Management	16	15	94%
Software Product Engineering	16	16	100%
Intergroup Coordination	12	12	100%
Peer Reviews	6	6	100%

for the workers of a project to participate actively in the development of a system.

RUP offers poor support in a number of key practices for the key process areas *Software Quality Assurance*, *Organization Process Focus*, and *Peer Reviews*.

In RUP, version 2001.00.03, a number of activities to ensure project quality have been added, including the elaboration of a Quality Assurance Plan. However, there are a number of gaps in handling the problems detected in reviews and audits, and in reporting the results to the project team. RUP does not describe how measurements are made to determine cost and schedule status of the SQA activities.

RUP proposes that the current status of the organization should be described in the activity Assess Current Organization, mainly with respect to its current process and improvement areas. However, RUP does not describe how the identified improvements are implemented, coordinated, or accomplished, including the development of improvement plans and the review of these plans, as CMM recommends; that explains the low coverage percentage reached in the key process area *Organization Process Focus*.

The key process areas *Subcontract Management* and *Training Program* are not supported at all by the Rational Unified Process. RUP considers these key process areas organizational responsibilities that are beyond the scope of the software process [21] and, therefore, need to be supported through other means by the organization. In the analyses presented in Tables 4 and 5, therefore, key process areas *Software Subcontract Management* and *Training Program* will be left out.

4.4.2 Key Practices Organized by Common Features

In Table 4, the key practices are organized by common features in each key process area. The objective is to describe the percentage of key practices by key process areas in terms of common features and to identify the common features that are not supported by the key process areas.

The common features are *Commitment to Perform* (CM), *Ability to Perform* (AB), *Activities Performed* (AC), *Measurement and Analysis* (MA), and *Verifying Implementation* (VI). The common features are briefly described in Section 3.

Table 4 shows the low percentage achieved by the common feature *Ability to Perform* in support of all key practices. The common feature *Ability to Perform* involves resources, organizational structure, and training.

Table 4 highlights that RUP focuses on software project management and software building processes, but it is not so keen on aspects related to systems management, such as cost management, training, human resource management, communications management, and procurement management, i.e., the precondition activities described by the common feature *Ability to Perform*. This is very natural and perfectly understandable because RUP evolves from the unification of methods for software development [4], [13], [25], and not from project management processes. Nevertheless, those indices show the need to examine the missing practices carefully.

RUP describes the purpose of Project Management Workflow as the provision of a framework for managing software-intensive projects, the provision of practical guidelines for planning, staffing, executing, and monitoring projects, and the provision of a framework for managing risk [21]. It is said that RUP does not attempt to cover all aspects of project management and that it does not cover issues such as managing people, managing budget, and managing contracts [21].

RUP suggests that procedures to satisfy these issues are out of its scope and must be defined by the organization [21]. The organization could use a specific approach for project management to complement RUP [21], such as the Project Management Institute's Project Management Body of Knowledge (PMBOK) [20]. PMBOK identifies specific knowledge areas to treat human resources management, procurement management, integration management, and

other management issues. Other authors examined RUP from a project management viewpoint [8], [30], [31].

Table 5 shows the percentages of key practices supported by key process area, eliminating practices described in the common feature *Ability to Perform*. This common feature was eliminated considering that the organizational preconditions are beyond the scope of RUP.

Analysis of this table shows that, when excluding the key practices of CMM described in the common feature *Ability to Perform*, the percentages supported by RUP in each key process area substantially increase, reaching 100 percent in many key process areas.

The key process area *Peer Review*, which had a percentage of 67 percent in Table 2, has its percentage risen to 100 percent in Table 5. This is because this KPA has nine key practices and three of them are related to preconditions (training and people) that are not considered in RUP.

The next section makes some propositions to extend the Rational Unified Process to conform to CMM.

4.4.3 Some Propositions to Conform to CMM

With the objective of extending RUP to satisfy CMM key practices, some proposals are suggested. These proposals describe the inclusion/alteration of activities, inclusion/alteration of artifacts, inclusion of workflows, and inclusion of workers or groups of workers.

Regarding activities, the propositions are to create activities *to estimate resources and funds, to estimate and track critical computational resources, to define activities that must be executed by software quality assurance group to ensure the software process steps and standards are followed*, and *to improve the organization's software process*. Regarding artifacts, RUP should also have documents to specify *project plans, process improvement plan*, and register *required resources at plans developed*. Finally, regarding workflow, a set of activities, artifacts, and workers to implement *Subcontract Management* and *Training* must be defined.

This paper will briefly describe only two examples of proposals. A more comprehensive set of propositions to bring RUP up to CMM levels 2 and 3 is described in Lisandra Manzoni's recently presented master dissertation [33].

Proposal 1: New procedures to estimate and track critical computer resources. In order to satisfy the key practices *Activities Performed 11/15* of the key process area *Software Project Planning*; *Activities Performed 7/13* of the key process area *Software Project Tracking and Oversight*, and *Activities Performed 8/11* of the key process area *Integrated Software Management*, procedures are required to estimate and track critical computer resources (computer memory, computer processor, communications channel capacity, etc.) needed by the project.

The key practice *Activities Performed 11/15* of the key process area *Software Project Planning* describes that "*Estimates for the project's critical computer resources are derived according to a documented procedure*."

The key practice *Activities Performed 7/13* of the key process area *Software Project Tracking and Oversight* describes that "*The project's critical computer resources are tracked, and corrective actions are taken as necessary*." The key practice *Activities Performed 8/11* of the key process area *Integrated Software Management* describes that "*The project's critical computer resources are managed according to a documented procedure*."

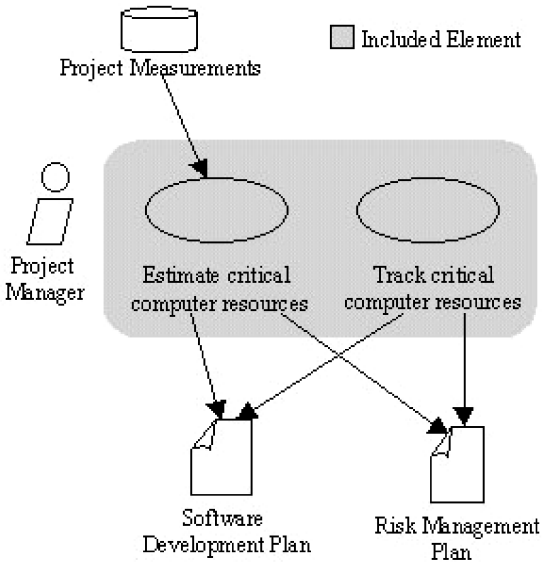

Fig. 1. Estimation and tracking of critical computer resources.

In order to estimate the critical resources, this procedure must use the historical information of similar projects. The estimated *critical computer resources* must be documented at the *software development plan*. Identified risks must be associated to critical computer resources in the *risk management plan*, to ensure that they will be monitored and controlled, as shown in Fig. 1.

The evaluation of the critical resources must be done at minor and major milestones, and corrective actions must be taken when risks arise in the project.

Proposal 2: Definition of a new workflow: Subcontract Management Workflow. RUP does not show a defined process with respect to subcontract management. The aim of this proposal is to formalize the subcontract management process and to satisfy the key process area *Subcontract Management*, through the definition of a new workflow, called Subcontract Management Workflow. This workflow is defined by using an activity diagram, as shown in Fig. 2.

This workflow proposes the inclusion of seven new activities, which are *develop a subcontract management plan, select subcontractor, elaborate contract and define commitments, track activities of the subcontractor, perform reviews and audits, monitor activities configuration management*, and *evaluate artifacts developed*. Included artifacts are *subcontract management plan* and *contract*, and the worker *subcontract manager*.

4.5 To Reach CMM Levels 2 and 3

An organization deploying RUP that desires to reach CMM level 2 needs to define its own procedures to select and to train human resources, to manage subcontracts, to provide funds to the project, to estimate and to track critical computer resources for the project, and to coordinate the software project planning with the overall project planning. It is also necessary to define how problems identified in the reviews must be handled and related to the workers, what measurements will be made to determine status of the SQA

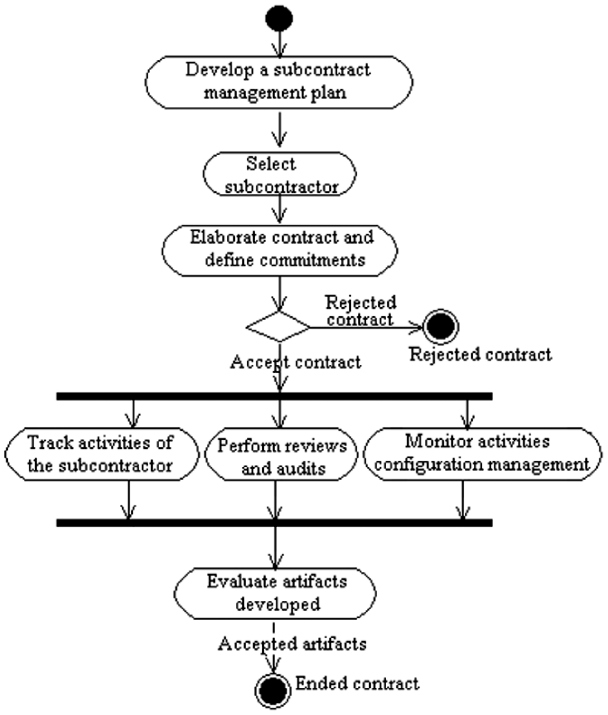

Fig. 2. Workflow of subcontract management.

activities, and how to include reviews of all performed activities to assure software quality.

In order to reach CMM level 3, it is necessary to describe how the software process of the organization can be improved, including the development of improvement plans and the review of these plans by senior management; and the organization must define procedures to identify the training needed by project team members, and then develop or procure training to address the identified needs.

5 Related Work

In [24], an evaluation of the Rational Unified Process in respect to standard ISO/IEC15504—Part 5 is described. Part 5 contains a detailed elaboration of mandatory processes and defines associated base practices, work products, and management practices, which are used to determine the capability for each assessed process. The report presents a summary as a graph of ratings against processes grouped by category and concludes that the ratings reflect the engineering focus of RUP, which fails to meet the requirements of standard 15504 at higher ratings in regards mainly to the management and organizational areas and the quality assurance functions.

RUP was also assessed in regards to its conformity to ISO 9000-3 in [1]. The authors concluded in this report that the implementation of RUP needs to be supported by complementary actions to satisfy the standard because RUP does not support several requirements of ISO 9000-3, such as definition of contracts, organizational responsibilities and policies, control of external products/services, etc.

In [8], the object-oriented processes RUP and OPEN (Object-Oriented Process, Environment, and Notation) are examined from a project management viewpoint and evaluated whether either or both would meet acceptable standards in process support, project management guidelines, and full lifecycle description for OO software development. The authors conclude that both processes are deficient in certain standard project management areas of knowledge to be fully self-contained processes capable of supporting the full suite of project management techniques and that therefore further extensions into these subdomains would seem to be desirable.

In [30], Hull et al. identify the best practices for software development and provide a basis for the assessment of three processes: Catalysis, OPEN, and RUP. The authors conclude that RUP is presented in a very management oriented format, whereas Catalysis is described as focusing on software engineering, while OPEN is somewhere in between the two. The level of detail of RUP is aimed at managers as opposed to software managers who would benefit more from descriptions and applications of UML. OPEN is a rich process and is documented clearly. Its level of detail is nearer that of Catalysis, being of use to both the manager and the software engineering. Catalysis is more detailed [30].

A qualitative evaluation is performed on the public domain component of RUP and on OPEN in [31]. The authors focus their comparison on aspects of the process architecture and underpinning metamodel, the concepts and terminology utilized and support for project management. The authors conclude that the metalevel architecture of RUP leads to some dilemmas in terms of the lack of support for a truly iterative development and an over-reliance on use-cases. OPEN combines the adaptability to construct a process to the specific needs of an individual domain and to adapt the process continually to particular projects. OPEN offers more extensive support in the area of cross-project suites of application developments and maintenance and it offers more extensive support in metrics and quality considerations. The authors concluded that these differences probably reflect the more underpinning model provided by developers of OPEN rather than the more pragmatic, tools-focused considerations of RUP developers [31].

In [27], an evaluation of how RUP meets the goals in each key process area of CMM levels 2 and 3 is described. The difference between that evaluation and the analysis reported in this paper is on focus. While [27] looked at 33 goals in CMM, this analysis looked into 229 key practices and their organization in common features. The Rational white paper [28] addresses how an organization can use RUP to achieve CMM level 2 key practices. Some differences in conclusions between this paper and Rational White Papers [27], [28] are as follows: The Rational White Papers state that the Software Development Plan defines the overall plan for the project, but this plan is concerned only with software planning. Some activities and work products required by CMM are lacking in RUP, and those papers do not address these issues clearly. Some of the missing activities involve cost management, human resources management (training), estimating critical computer resources, handling of problems detected in reviews and audits, and continuous software process improvement. This paper goes further, pointing to missing key practices,

quantifying the coverage that RUP offers by key process area and suggesting extensions to missing key practices.

6 Summary and Conclusion

The assessment shows that the Rational Unified Process meets most of the Capability Maturity Model requirements, but some of the more managerial aspects of the software process are not currently supported.

The Rational Unified Process presents a well-defined approach on software project management and software engineering processes, but it is not an approach centered on systems management concerns. Therefore, it lacks activities involving issues such as cost management, human resource management, communications management, and contracts management.

CMM key process areas better supported by RUP are Requirements Management, Software Project Planning, Software Project Tracking and Oversight, Software Configuration Management, Organization Process Definition, and Software Product Engineering. RUP offers good support to key process areas Integrated Software Management, and Intergroup Coordination. Key process areas Software Quality Assurance, Organization Process Focus, and Peer Reviews show low support to the corresponding key practices recommended by CMM. RUP does not support key process areas Software Subcontract Management and Training.

An organization aiming at CMM levels 2 and 3 must complement RUP. It must develop its own procedures to satisfy key practices not supported by RUP, or use a specific project management approach to deal with human resource management, budget management, and contract management.

This study was started at the beginning of 2000 and was first conducted on the RUP available then (version 5.5) [22]. RUP has considerably improved its coverage of CMM practices from version 5.5 to version 2001. From a key process area coverage of only 35 percent (minimum) to 79 percent (maximum) for CMM key practices in version 5.5, it now covers 44 percent (minimum) to 86 percent (maximum) in its 2001 version. The improvements have come from the emphasis the new version places on the definition of procedures for software quality assurance. With the inclusion of software quality assurance activities, RUP satisfies the key practices of the common features Verifying Implementation, included in almost all key process areas, thus increasing the percentage of coverage of key practices in these KPAs.

This paper identifies missing key practices and proposes some activities and artifacts to complement RUP. The authors are working on the elaboration of templates for the proposed artifacts and on guidelines to the proposed activities.

Some results described in this assessment have also been evaluated by the use of the Rational Unified Process in real projects, mainly within the Brazilian Federal Data Processing Agency SERPRO. The authors would like to express their gratitude to the SERPRO managers that allowed interaction to happen.

Acknowledgments

This project was partially supported by the Brazilian Research Agency CNPq under the grant Amadeii 5228-2519-99/8.

References

[1] A.A. Alcântara, T.M.M. Maciel, S.L. Meira, and F.Q.B. da Silva, "Uso do Processo RUP na Implantação da ISO 9000-3," *Proc. Sixth Workshop Qualidade de Software,* pp. 60-71, 1999.

[2] A. April and F. Coallier, "Trillium: Model for Telecom Product Development and Support Process Capability," *Proc. Second IEEE Software Eng. Standards Symp.,* Aug. 1995.

[3] J. Boegh, S. Depanfilis, B. Kitchenham, and A. Pasquini, "A Method For Software Quality Planning, Control, and Evaluation," *IEEE Software,* vol. 16, no. 2, pp. 69-85, Mar. 1999.

[4] G. Booch, *Object-Oriented Analysis and Design with Applications,* second ed. The Benjamin/Cummings Publishing Co., 1994.

[5] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language Users Guide.* Addison Wesley, 1999.

[6] B. Fitzgerald and T. O'Kane, "A Longitudinal Study of Software Process Improvement," *IEEE Software,* vol. 16, no. 3, pp. 37-45, May 1999.

[7] V. Haase, G. Koch, H.J. Kugler, and P. Decrinis, "Bootstrap: Fine-Tuning Process Assessment," *IEEE Software,* vol. 11, no. 4, pp. 25-35, July 1994.

[8] B. Henderson-Sellers, R. Dué, and I. Graham, "Third Generation OO Processes: A Critique of RUP and OPEN from a Project Management Perspective," *Proc. Seventh Asia-Pacific Software Eng. Conf. (APSEC' 2000),* pp. 428-435, 2000.

[9] J. Herbsleb, A. Carleton, J. Rozum, J. Siegel, and D. Zubrow, "Benefits of CMM-Based Software Process Improvement Initial Results," Technical Report CMM/SEI-94-TR-13, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Penn., 1994, http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr13.94.pdf.

[10] ISO 9000-3, "ISO 9000-3 Guidelines for the Application of ISO 9001 to the Development, Supply, and Maintenance of Software," Int'l Standards Organization, Geneva, 1991, http://www.iso.org/.

[11] ISO/IEC 12207, "ISO/IEC 12207 Information Technology—Software Life-Cycle Processes," Int'l Standards Organization, Geneva, 1995, http://www.iso.org/.

[12] I. Jacobson, G. Booch, and J. Rumbaugh, *Unified Software Development Process.* Addison-Wesley, 1998.

[13] I. Jacobson, M. Christerson, P. Jonsson, and G. Overgaard, *Object-Oriented Software Engineering: A Use Case-Driven Approach.* Addison-Wesley, 1992.

[14] P. Kruchten, *The Rational Unified Process—An Introduction.* Addison-Wesley, 2000.

[15] E.G. McGuire, "Initial Effects of Software Process Improvement on an Experienced Software Development Team," *Proc. 29th Ann. Hawaii Int'l Conf. System Sciences,* pp. 713-721, 1998.

[16] M. Paulk, "Effective CMM-Based Process Improvement," *Proc. Sixth Int'l Conf. Software Quality (ICSQ'96),* pp. 226-237, 1996.

[17] M. Paulk, B. Curtis, M. Chrissis, and C. Weber, "Capability Maturity Model for Software," Version 1.1, Technical Report CMU/SEI-93-TR-24, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Penn., 1993, http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr24.93.pdf.

[18] M. Paulk et al., "Key Practices of Capability Maturity Model for Software," Version 1.1, Technical Report CMU/SEI-93-TR-25, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Penn., 1993, http://www.sei.cmu.edu/pub/documents/93.reports/pdf/tr25.93.pdf.

[19] CMMI Product Team, "Capability Maturity Model, Integration (CMMI)," Version 1.1, Technical Report CMU/SEI-2002-TR-011, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Penn., Mar. 2002, http://www.sei.cmu.edu/pub/documents/02.reports/pdf/02tr011.pdf.

[20] Project Management Institute, Inc., "A Guide to the Project Management Body of Knowledge," 2000 ed., 2000, http://www.pmi.org.

[21] Rational Software Corporation, "Rational Unified Process," Version 2001.3, CD-ROM, Rational Software, Cupertino, Calif.: 2001.

[22] Rational Software Corporation, "Rational Unified Process,"Version 5.5, CD-ROM, Rational Software, Cupertino, Calif.: 1999.

[23] Rational Software Corporation, "Rational Unified Process Best Practices for Software Development Teams," White Paper, 1998, http://www.rational.com/media/whitepapers/rup_bestpractices.pdf.
[24] Rational Software Corporation, "Assessing the Rational Unified Process against ISO/IEC15504-5: Information Technology—Software Process Assessment Part 5: An Assessment Model and Indicator Guidance," White Paper, 2000, http://www.rational.com/products/whitepapers/rup_iso.jsp.
[25] J. Rumbaugh, M.R. Blaha, W. Lorensen, F. Eddy, and W. Premerlani, *Object-Oriented Modeling and Design.* Prentice-Hall, 1991.
[26] ISO/IEC 15504, "Information Technology—Software Process Assessment—Part 1: Concepts and Introductory Guide," Technical Report ISO/IEC TR 15504-1:1998, Int'l Standards Organization, JTC 1/SC 7, 1998, http://www.iso.org/.
[27] J. Smith, "Reaching CMM Level 2 and 3 with the Rational Unified Process," White Paper, 2000, http://www.rational.com/products/whitepapers/100416.jsp.
[28] R.W. Reitzig, C. Rodriguez, and G. Holt, "Achieving Capability Maturity Model Level 2 with Rational Unified Process," Rational White Paper, 2002, http://www.cognence.com/AchievingCMM-Level2.pdf.
[29] M.P. Ginsberg and L.H. Quinn, "Process Tailoring and the Software Capability Maturity Model," Technical Report CMU/SEI-94-TR-024, Software Eng. Inst., Carnegie Mellon Univ., Pittsburgh, Penn. 1995, http://www.sei.cmu.edu/pub/documents/94.reports/pdf/tr24.94.pdf.
[30] M.E.C. Hull, P.S. Taylor, J.R.P. Hanna, and R.J. Millar, "Software Development Processes—An Assessment," *Information and Software Technology,* vol. 44, no. 1, pp. 1-12, Jan. 2002.
[31] B. Henderson-Sellers, G. Collins, R. Dué, and I.M. Graham, "A Qualitative Comparison of Two Processes for Object-Oriented Software Development," *Information and Software Technology,* vol. 43, no. 12, pp. 705-724, Dec. 2001.
[32] A. Fuggeta, "Software Process: A Roadmap," *Proc. 22nd Int'l Conf. Software Eng.—Future of Software Eng.,* pp. 25-34, 2000.
[33] L.V. Manzoni, "Using a Workflow Management System to Support Software Development Based on Extended Rational Unified Process to Reach Maturity Model Levels 2 and 3," master dissertation, Inst. of Informatics, Federal Univ. of Rio Grande do Sul, Porto Alegre, Brazil, 2001, http://www.inf.ufrgs.br/amadeus/atuais/lisandra.html.

Lisandra V. Manzoni received the MSc degree in computer science from the Universidade Federal do Rio Grande do Sul (UFRGS) and the BSc degree from the Universidade Federal de Santa Maria (UFSM). She is currently a PhD student in computer science at UFRGS. She works at Serviço Federal de Processamento de Dados in research and software development. Her research interests are on software process, software project management, and tools. She is member of the Brazilian Computing Society.

Roberto T. Price received the PhD degree from the University of Sussex-UK (School of Engineering and Applied Sciences), the MSc degree from the University of London (LSE), and the BSc degree from the Universidade Federal do Rio Grande do Sul (Escola de Engenharia). He is currently VP of R&D at TeleNova Communications and also a collaborator-professor at the Instituto de Informática at the Universidade Federal do Rio Grande do Sul, where he had a career as researcher, lecturer, and academic administrator. His research interests in Software Engineering are on tools, methods, and processes. Dr. Price is a founding member of the Brazilian Computing Society. He is a member of the IEEE and the IEEE Computer Society.